by Charles Petzold

Environments

An Introduction to Rendering Color on Video Displays

■ Color is vitally important to our interpretation of visual information. But in order to work with color on computers, we need to find a way to quantify it, first.

The human eye is sensitive to electromagnetic radiation with wavelengths ranging from 400 to 700 nanometers (nm). This is known as the visual spectrum of light. The different wavelengths of visible light affect our eyes in different ways, which we interpret as color.

Light from the sun includes radiation from the entire range of 400 to 700 nm in approximately equal amounts. We call this white light. Passing the light through a prism allows us to see the various colors of light within this range. These are the colors of the rainbow, traditionally labeled as red (at 700 nm), orange, yellow, green, blue, indigo, and violet (at 400 nm).

Color is vitally important to our interpretation of visual information. Hence, if we wish to extend the use of the computer into rendering visual information on a video display or other output device through the use of computer graphics, we must have a firm grasp of color.

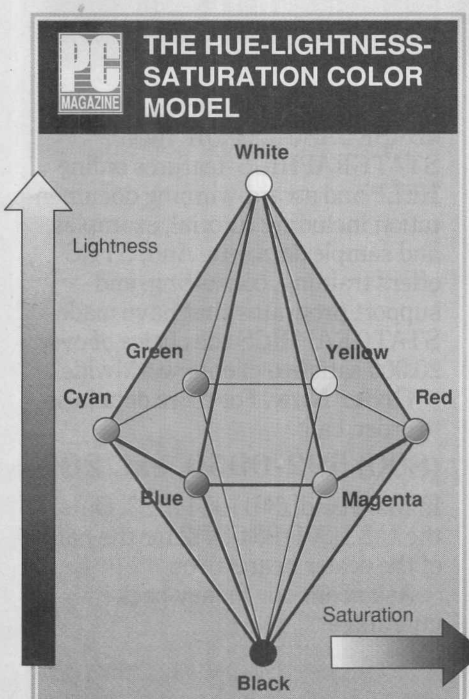


Figure 1: This is the Hue-Lightness-Saturation (HLS) double-hexcone color model. Note that the hues form a circle of color around the middle.

In the next several Environments columns, I'll discuss color, and more specifically, how it is handled in the OS/2 Graphics Programming Interface (GPI). We'll also explore the new palette manager found in OS/2 2.0.

In this issue, I'll examine different ways of quantifying color, and also some of the common color video display hardware available for IBM-compatible personal computers.

REPRESENTATIONS OF COLOR

Computers deal with numbers, so in order to work with color on computers, first we need to find some way to quantify it. This involves both giving the user of a computer program a way to select a color and allowing the program some way to specify the color when rendering it on an output device.

There are several different ways to do this, and they're all somewhat difficult to visualize, because they involve three parameters. Consequently, trying to represent the entire range of color on a two-dimensional surface is rather awkward.

Two approaches to selecting color that users of a computer program usually find most natural are called the *Hue-Lightness-Saturation* (HLS) model and the *Hue-Saturation-Value* (HSV) model. These two models are closely related—the lightness parameter of HLS is equivalent to the value parameter of HSV.

The HLS model is used in the OS/2 Control Panel program for selecting system colors. The user selects a color by moving three scrollbars. In the Control Panel, the scrollbars are labeled *Color, Shade,* and *Amount,* which correspond respectively with *Hue, Lightness,* and *Saturation* in the HLS model.

The HLS model is generally visualized as a double-hexcone, as shown in Figure 1.

The *hue* is the dominant wavelength of the light. The range of hues can be derived from the colors of the rainbow, since each of the colors corresponds to a particular wavelength. In doing this, normally the colors are relabeled somewhat, with some parts of the spectrum compressed and other parts stretched out to yield the following:

```
Red
Yellow
Green
Cyan
Blue
Magenta
Red
```

You'll note that I've added red to the bottom, indicating that these hues really

PC MAGAZINE
EDITOR'S CHOICE
STATGRAPHICS
March 14, 1989

NEW! Automated Processing. Upgrades Available

Expert statistics software. For people who aren't statistics experts.

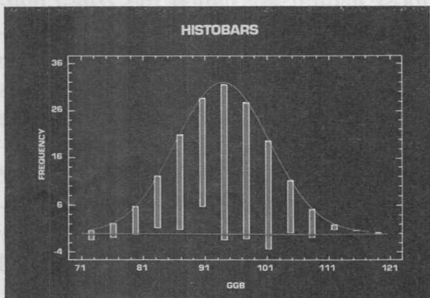
Most PC statistics packages were designed for statistics experts. Which means to use them, you need to memorize obscure algorithms and cryptic commands. That takes time.

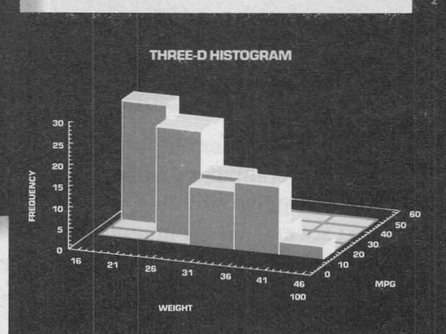
Only one PC statistics package is powerful, comprehensive *and* easy to use: STATGRAPHICS®.

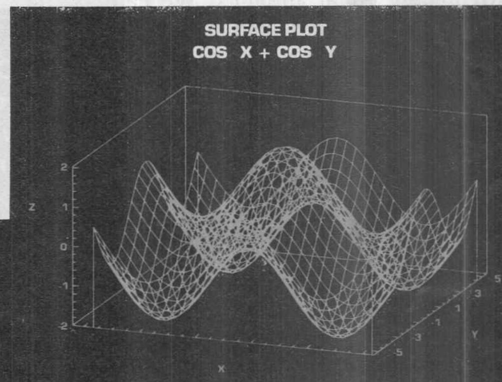
Comprehensive Statistics. STATGRAPHICS includes over 250 statistical procedures. All the tools you need for complete statistical analysis: ANOVA, complete regression analysis, multivariate and nonparametric techniques, quality control and experimental design, exploratory data analysis, extensive forecasting and time series analysis, and more.

Integrated Statistical Graphics. Visualize your data instantly with presentation graphics completely integrated with your statistics—3-D histograms, quality control charts, 2 and 3-D line and surface plots, bar and pie charts, and more.

Modify data and assumptions repeatedly, query data points, do on-screen forecasting and model fitting, overlay graphs, or zoom in on any area for a closer look—without ever leaving the procedure or making permanent changes to your data.







This unique interactive environment lets you test ideas more quickly, analyze data more thoroughly, and uncover hidden trends.

Easy To Use. Enter data using STATGRAPHICS full-screen editor. Or import it directly from Lotus®, dBASE®, ASCII, or DIF files. STATGRAPHICS features online HELP and award-winning documentation including tutorial, examples, and sample data sets. And, STSC offers training, consulting, and support programs that have made STATGRAPHICS the choice of over 20,000 satisfied clients worldwide.

STATGRAPHICS–The Best Overall Value!

Product	Completely Menu Driven	Interactive Graphics	Storage Required	U.S. Sugg. Price
STATGRAPHICS	✔	✔	2 meg.	$895 Complete
SPSS™/PC+	No	No	10 meg.	$2375*
SAS®/PC	No	No	20 meg.	$2075*

Order Now. For more details or to order, call:

(800) 592-0050, ext. 200.

In Maryland, (301) 984-5123. Outside the U.S., (301) 984-5412 for the name of the dealer nearest you.

Ask about our money-back guarantee!

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, MD 20852

CIRCLE 365 ON READER SERVICE CARD

STATGRAPHICS is a registered trademark of Statistical Graphics Corporation. Other trademarks are the property of their respective owners. International prices may vary.
*Price includes base system plus all modules necessary to fully emulate STATGRAPHICS. Chart based on data compiled as of 8/2/89.

Environments

form a circle of color rather than a straight line, as shown at the middle of the diagram in Figure 1. Generally, the hues are represented by angles, beginning with red at zero degrees and moving around the circle in a counterclockwise direction.

The *saturation* parameter reflects the purity of the color. At its minimum value of zero, the color is a shade of gray, indicating that the color combines wavelengths along the entire visual spectrum. The vertical line in Figure 1 represents the *lightness* parameter. A lightness value of zero yields black and a maximum lightness value yields white, with all the shades of gray ranged in between.

RGB AND CMY

When rendering color on a graphics output device such as a video display or a color printer, two more color models become useful. These two color models are each based on three primary colors, but their primary colors are different.

The additive primary colors are red, green, and blue. These additive primaries are used with graphics output devices that render color using a light source, such as a video display. All possible colors that a video display can render are composed of various combinations of red, green, and blue. (We'll see shortly that this is exactly how video displays work.)

The Red-Green-Blue (RGB) color model is generally pictured as a three-dimensional coordinate system or a cube, as shown in Figure 2.

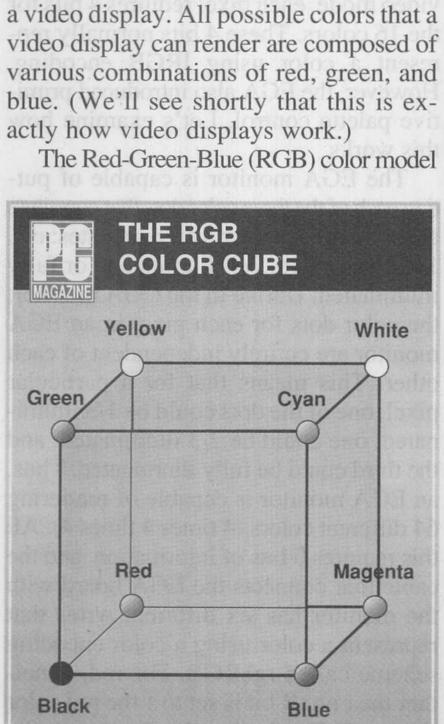


Figure 2: The Red-Green-Blue (RGB) color model is generally pictured as a three-dimensional coordinate system or a cube.

Black is at the lower-left-forward corner. This is the origin of a three-dimensional coordinate space. Increasing values of blue are to the right, increasing values of green are toward the top, and increasing values of red are toward the back.

The combination of blue and green is cyan, the combination of blue and red is magenta, and the combination of red and green is yellow. These three additional colors are at three other corners of the cube. The upper-right-back corner is the combination of red, green, and blue, or white. The interior diagonal from the black corner to the white corner represents all the shades of gray.

When rendering color on white paper, a different color model is required. The paper reflects white light, so the inks on

the paper must subtract various colors of light from the white light. The colors that do this are cyan, magenta, and yellow, and are known as the subtractive primaries. This is the CMY color model. It can be pictured the same way as the RGB color model, except with white as the origin instead of black. With subtractive primaries, the combination of cyan and magenta is blue, the combination of cyan and yellow is green, and the combination of magenta and yellow is red.

Note that the three additive primaries and the three subtractive primaries are all shown in the HLS color model in Figure 1. The additive primaries and subtractive primaries alternate, with the combination of each pair yielding the color in between.

The RGB color model is ideal for specifying color on a video display.

Algorithms for converting between these various color models can be found in *Computer Graphics: Principles and Practice* by Foley, van Dam, Feiner, and Hughes (Addison-Wesley, 1990). This is the second edition of one of the true classics in the field of computer graphics. I heartily recommend it.

COLOR VIDEO HARDWARE

For the remainder of this discussion, I'll concentrate on rendering color on video displays rather than on color printers.

The RGB color model is ideal for specifying color on a video display, because it is precisely the way that video displays render color. A color monitor has three electron guns: one for red, one for green, and one for blue. Each pixel displayed on the monitor is a combination of a red, a green, and a blue dot.

Let's first assume we have a rather primitive color video board and monitor combination. With this primitive hardware, the red, green, and blue dots that define the color of the pixel can only be in one of two states: on (that is, illuminated) or off.

The memory on the video board would require 3 bits for each pixel. A 1 bit indicates that the dot is on, and a 0 bit indicates that the dot is off. The various combinations allow rendering eight different colors, as follows:

R	G	B	Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

But there's a little practical problem with this. There are 8 bits in a byte, and 8 is not equally divisible by 3 (the number of bits per pixel). If each byte of video memory corresponds to two pixels, then 2 bits per byte are being wasted. If we organize video memory so that no bits are wasted, then pixels would straddle byte boundaries.

IRGB COLOR ENCODING

In order to avoid this problem, the old IBM Color Graphics Adapter (CGA) and monitor introduced a fourth bit, which came to be known as the Intensity bit. This color encoding scheme was called Intensity-Red-Green-Blue (IRGB), and is shown below:

I	R	G	B	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	Light gray
1	0	0	0	Dark gray
1	0	0	1	Light blue
1	0	1	0	Light green
1	0	1	1	Light cyan
1	1	0	0	Light red
1	1	0	1	Light magenta
1	1	1	0	Yellow
1	1	1	1	White

With the CGA monitor, each color dot is capable of being in one of three different states: off, half-illuminated, or fully illuminated. Normally a monitor that was capable of three states for each color do would be able to render 27 different colors (3 times 3 times 3). However, on the CGA monitor it is not possible for any particular pixel to have some half-illuminated dots and some fully illuminated dots. That's why the number of colors is limited to 16.

Of course, this IRGB color encoding is familiar to anyone who has programmed for the CGA or later IBM video adapters under DOS in character mode. IRGB color encoding is used for specifying the foreground and background colors of characters in the attribute byte.

The IRGB color encoding scheme may seem like a mere historical curiosity if you're doing device-independent graphics programming under the OS/2 Presentation Manager, but we'll see in the next issue that IRGB still makes its presence known in a very strong way.

ENHANCED GRAPHICS

IBM's next video adapter was known as the Enhanced Graphics Adapter (EGA). The most significant enhancement was the introduction of a graphics video mode of 640 by 350 pixels with 16 differen colors. With the EGA in this graphics video mode, each pixel requires 4 bits for the 16 colors. These 4 bits normally represent a color using IRGB encoding. However, the EGA also introduced primitive palette control. Let's examine how this works.

The EGA monitor is capable of putting each of the three color dots that constitute a pixel in one of four different states: off, 1/3 illuminated, 2/3 illuminated, or fully illuminated. Unlike in the CGA monitor, the color dots for each pixel in an EGA monitor are entirely independent of each other. This means that for a particular pixel, one of the dots could be 1/3 illuminated, one could be 2/3 illuminated, and the third could be fully illuminated. Thus, an EGA monitor is capable of rendering 64 different colors (4 times 4 times 4). All this requires 6 bits of information, and the cable that connects the EGA board with the monitor has six different wires that represent a color using a color encoding scheme called rgbRGB. For red, if neither the r nor R bit is set to 1 the red color dot is off. If only the r bit is set to 1, the color dot is 1/3 illuminated. If only the R bit is set to 1, the color dot is 2/3 illuminated. If both the r and R bits are set to 1,

the color dot is fully illuminated.

However, the EGA video board memory is organized to allow only 4 bits per pixel. The 4 pixel bits are translated into 6-bit rgbRGB colors, which are then sent to the monitor. This translation is accomplished through use of a ''palette table,'' which is simply a lookup array with 16 6-bit entries.

The EGA BIOS loads rgbRGB values into the palette table that cause the EGA to function as if it used IRGB color encoding. However, a program using the EGA can load different values into the palette table so that the 4 bits used to specify a particular color have nothing to do with IRGB encoding.

Therefore we say that the EGA is capable of 16 simultaneous colors chosen from a palette of 64 colors.

The number of simultaneous colors a video adapter is capable of rendering is determined by the amount and organization of memory on the video board—specifically, how many bits are available for each pixel. The number of colors in the full palette is governed by the video board's palette registers and the interface between the board and the monitor.

Up to this point, all the hardware we've examined has involved digital signals (either on or off) that are sent from the video board to the monitor. Thus, the 64 different colors that the EGA monitor is capable of producing require six wires in the cable for the purpose of color control. Each pair of wires governs the four different states of one of the three color dots that comprise a pixel.

ANALOG COLOR HARDWARE

With the introduction of the PS/2 series in 1987, IBM moved to analog monitors. These monitors require only three wires in the cable for color control—one wire for red, one for green, and one for blue. The monitor can respond to different signal levels in these wires for the intensities of the three color dots. Thus, at least in theory, these analog monitors are capable of an infinite range of color.

With the PS/2 series, IBM also introduced the Video Graphics Array (VGA). When used in the 640-by-480 resolution required by the OS/2 Presentation Manager, color is specified using only 4 bits per pixel. Thus, the adapter is capable of 16 simultaneous colors.

The number of simultaneous colors a video adapter can render is determined by the organization and amount of memory on the video board.

Some Super VGA boards can do better than that. These boards can switch into high-resolution video modes that allow using 8 bits per pixel for 256 simultaneous colors. IBM's 8514/A (when fully populated with memory) also allows using 8 bits per pixel.

The VGA (and other analog boards) also have more-extensive palette control than the EGA. Rather than the 4 pixel-bits mapping to a 6-bit color value, the 4 pixel-bits of the VGA (or the 8 pixel-bits of some Super VGAs and the 8514/A) map to an 18-bit color value. This means there are 6 bits for red, 6 bits for green, and 6 bits for blue. These 6-bit values pass through a digital-to-analog converter (DAC) to control the levels of the red, green, and blue signals sent to the monitor, as shown in Figure 3.

The use of 18 bits for deriving the color signals results in a palette of 262,144 colors (2^{18}). Thus we say that the VGA is capable of rendering 16 simultaneous colors chosen from a palette of 262,144 colors. The 8514/A and some Super-VGA boards are capable of 256 simultaneous colors, also chosen from a palette of 262,144 colors.

PIXEL-BITS AND COLOR

For the EGA and later adapters, the pixel-bits stored in video board memory mean nothing by themselves. They are merely an index into the palette lookup table. The value in the palette table is what determines the actual color associated with each unique pixel-bit value.

For example, on the VGA, a pixel-bit value of 0000 could mean red, 0001 could be light red, 0010 could mean black, 0011 could mean white, and so forth. However, as we'll see in a future column, the video display driver for the OS/2 Presentation Manager configures the VGA so that the pixel-bit values conform to the traditional IRGB encoding.

So what? Does it really matter to a PM program how the pixel-bit values map to actual colors in the palette table? After all, a PM program doesn't get anywhere near using pixel-bit values. That's all handled by the device driver.

As it turns out, the configuration of the palette table has important implications for working with raster operations and mix modes. That's why it's important to understand what's going on at the level of the hardware.

As we'll also see in future columns, OS/2 1.2 has, in theory, a facility to allow a PM program to change the palette table in the video adapter—but this has never been implemented. OS/2 2.0 adds functions that give a program extensive control over the palette. ■

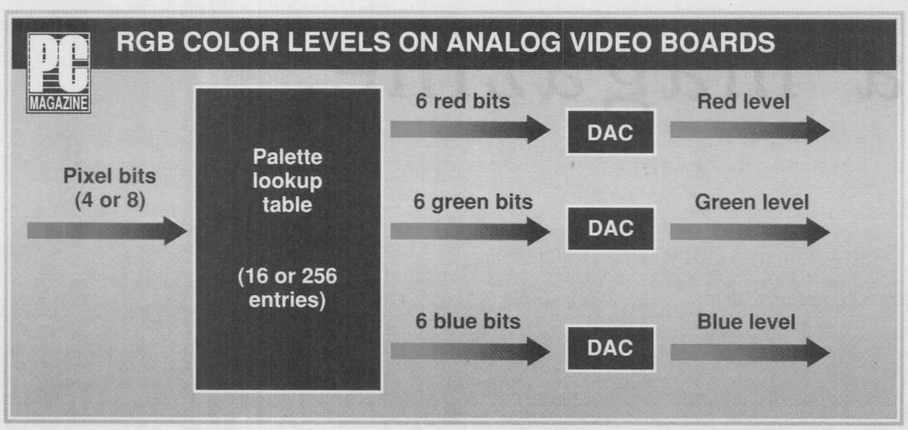


Figure 3: The mapping of pixel-bits to RGB color levels on VGA, Super VGA, and 8514/A video boards.